



AF

JFW

TRANSMITTAL FORM <i>(to be used for all correspondence after initial filing)</i>		Application No.	09/821,640
		Filing Date	March 29, 2001
		First Named Inventor	Guei-Yuan Lueh
		Art Unit	2193
		Examiner Name	Kang, Insun
Total Number of Pages in This Submission	31	Attorney Docket Number	42390P9758

ENCLOSURES (check all that apply)		
<input checked="" type="checkbox"/> Fee Transmittal Form	<input type="checkbox"/> Drawing(s)	<input type="checkbox"/> After Allowance Communication to TC
<input checked="" type="checkbox"/> Fee Attached	<input type="checkbox"/> Licensing-related Papers	<input checked="" type="checkbox"/> Appeal Communication to Board of Appeals and Interferences
<input type="checkbox"/> Amendment / Reply	<input type="checkbox"/> Petition	<input type="checkbox"/> Appeal Communication to TC (Appeal Notice, Brief, Reply Brief)
<input type="checkbox"/> After Final	<input type="checkbox"/> Petition to Convert a Provisional Application	<input type="checkbox"/> Proprietary Information
<input type="checkbox"/> Affidavits/declaration(s)	<input type="checkbox"/> Power of Attorney, Revocation Change of Correspondence Address	<input type="checkbox"/> Status Letter
<input type="checkbox"/> Extension of Time Request	<input type="checkbox"/> Terminal Disclaimer	<input checked="" type="checkbox"/> Other Enclosure(s) (please identify below):
<input type="checkbox"/> Express Abandonment Request	<input type="checkbox"/> Request for Refund	<div>- First Class Certificate of mailing; and - Stamped return postcard</div>
<input type="checkbox"/> Information Disclosure Statement	<input type="checkbox"/> CD, Number of CD(s)	
<input type="checkbox"/> PTO/SB/08	<input type="checkbox"/> Landscape Table on CD	
<input type="checkbox"/> Certified Copy of Priority Document(s)	Remarks	
<input type="checkbox"/> Response to Missing Parts/Incomplete Application		
<input type="checkbox"/> Basic Filing Fee		
<input type="checkbox"/> Declaration/POA		
<input type="checkbox"/> Response to Missing Parts under 37 CFR 1.52 or 1.53		

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT	
Firm or Individual name	Aslam A. Jaffery, Reg. No. 51,841 BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
Signature	
Date	June 21, 2007

CERTIFICATE OF MAILING/TRANSMISSION			
I hereby certify that this correspondence is being deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to: Mail Stop Appeal Brief-Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.			
Typed or printed name	Krista Mathieson		
Signature		Date	June 21, 2007



FEE TRANSMITTAL for FY 2006

Patent fees are subject to annual revision.

Complete if Known

Application Number	09/821,640
Filing Date	March 29, 2001
First Named Inventor	Guei-Yuan Lueh
Examiner Name	Kang, Insun
Art Unit	2193
Attorney Docket No.	42390P9758

☐ Applicant claims small entity status. See 37 CFR 1.27.

TOTAL AMOUNT OF PAYMENT (\$) 500.00

METHOD OF PAYMENT (check all that apply)

☒ Check ☐ Credit card ☐ Money Order ☐ None ☐ Other (please identify): _____

☒ Deposit Account Deposit Account Number: 02-2666 Deposit Account Name: Blakely, Sokoloff, Taylor & Zafman LLP

For the above-identified deposit account, the Director is hereby authorized to: (check all that apply)

☐ Charge fee(s) indicated below

☒ Credit any overpayments

☐ Charge fee(s) indicated below, except for the filing fee

☒ Any concurrent or future reply that requires a petition for extension of time should be treated as incorporating an appropriate petition for extension of time and all required fees should be charged.

☒ Charge any additional fee(s) or underpayment of fee(s) during the pendency of this application.

FEE CALCULATION

1. EXTRA CLAIM FEES

Total Claims	Extra Claims	Fee from below	Fee Paid
24	29*	0	\$0.00
Independent Claims	4	5*	0
Multiple Dependent			

Large Entity Small Entity

Fee Code	Fee (\$)	Fee Code	Fee (\$)	Fee Description
1202	50	2202	25	Claims in excess of 20
1201	200	2201	100	Independent claims in excess of 3
1203	360	2203	180	Multiple Dependent claim, if not paid
1204	790	2204	395	**Reissue independent claims over original patent
1205	300	2205	150	**Reissue claims in excess of 20 and over original patent

**or number previously paid, if greater. For Reissues, see below

SUBTOTAL (1)

(\$) 0.00

2. ADDITIONAL FEES

Large Entity Small Entity

Fee Code	Fee (\$)	Fee Code	Fee (\$)	Fee Description
1051	130	2051	65	Surcharge - late filing fee or oath
1052	50	2052	25	Surcharge - late provisional filing fee or cover sheet.
2053	130	2053	130	Non-English specification
1251	120	2251	60	Extension for reply within first month
1252	450	2252	225	Extension for reply within second month
1253	1,020	2253	510	Extension for reply within third month
1254	1,590	2254	795	Extension for reply within fourth month
1255	2,160	2255	1,080	Extension for reply within fifth month
1401	500	2401	250	Notice of Appeal
1402	500	2402	250	Filing a brief in support of an appeal
1403	1,000	2403	500	Request for oral hearing
1451	1,510	2451	1,510	Petition to institute a public use proceeding
1460	130	2460	130	Petitions to the Commissioner
1807	50	1807	50	Processing fee under 37 CFR 1.17(q)
1806	180	1806	180	Submission of Information Disclosure Stmt
1809	790	1809	395	Filing a submission after final rejection (37 CFR § 1.129(a))
1810	790	2810	395	For each additional invention to be examined (37 CFR § 1.129(b))

Other fee (specify) _____

SUBTOTAL (2)

(\$) 500.00

Fee Paid

500.00

500.00

SUBMITTED BY

Complete (if applicable)

Name (Print/Type) Aslam A. Jaffery

Registration No.
(Attorney/Agent)

51,841

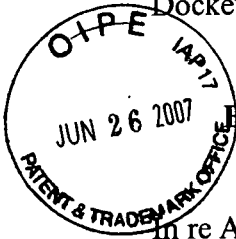
Telephone

(303) 740-1980

Signature

Date

06/21/07



**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

In re Application of:

Guei-Yuan Lueh, et al.)	Examiner:	Kang, Insun
Application No: 09/821,640)	Art Unit:	2193
Filed: March 29, 2001)		
For: Method and System of Controlling)		
Dynamically Compiled Native)		
Code Size)		

Mail Stop: Appeal Brief - Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF
IN SUPPORT OF APPELLANT'S APPEAL
TO THE BOARD OF PATENT APPEALS AND INTERFERENCES

Sir:

Applicant (hereafter "Appellant") hereby submits this Appeal Brief (hereinafter "Brief") in triplicate in support of its appeal from a final decision by the Examiner, mailed February 05, 2007 in the above-referenced Patent Application (hereinafter "Application). Appellant respectfully requests consideration of this appeal by the Board of Patent Appeals and Interferences (hereinafter "Board") for allowance of the above-captioned Application.

An oral hearing is not desired.

06/26/2007 CNEGA1 00000059 09821640

01 FC:1402

500.00 0P

TABLE OF CONTENTS

I.	REAL PARTY IN INTEREST.....	3
II.	RELATED APPEALS AND INTERFERENCES.....	3
III.	STATUS OF THE CLAIMS.....	3
IV.	STATUS OF AMENDMENTS.....	3
V.	SUMMARY OF THE CLAIMED SUBJECT MATTER.....	4
VI.	GROUND S OF REJECTION.....	5
VII.	ARGUMENT.....	6
VIII.	CONCLUSION.....	7
IX.	CLAIMS APPENDIX	i
X.	EVIDENCE APPENDIX	x
XI.	RELATED PROCEEDINGS APPENDIX	xi

I. REAL PARTY IN INTEREST

The invention is assigned to Intel Corporation of 2200 Mission College Boulevard, Santa Clara, California 95052-8119.

II. RELATED APPEALS AND INTERFERENCES

To the best of Appellant's knowledge, there are no appeals or interferences that are related to, will directly affect, will be directly affected by, or have a bearing on the Board's decision in the present appeal.

III. STATUS OF THE CLAIMS

Claims 1-15 and 17-24 are currently pending in the above-referenced Application. No claims have been allowed. All pending claims were rejected in the final Office Action mailed February 5, 2007 (hereinafter "Office Action"), and are the subject of this appeal.

IV. STATUS OF AMENDMENTS

Claims 1-15 and 17-24 are currently pending in the subject Application. These claims were finally rejected in the Office Action.

In response to the Office Action rejecting claims 1-15 and 17-24 under 35 U.S.C. §§102(e), Appellant filed a timely Notice of Appeal on May 4, 2007.

A copy of all claims on appeal is attached hereto as the Appendix of Claims.

Appellant respectfully traverses each of the grounds of rejection

V. SUMMARY OF THE CLAIMED SUBJECT MATTER

In one embodiment, a method is disclosed. The method comprises storing native code associated with a first method and a second method within a native code space, and associating a reference to the first method and the second method in a method table. The method includes determining whether the first method or the second method is invoked by detecting whether the first method or the second method correspond to the native code, and determining whether the native code space exceeds a threshold in response to an invocation of the first method or the second method. The method further includes incrementing method counters each time the first method or the second method is invoked, wherein the method counters correspond to the first method and the second method. The method also includes unwinding a stack to determine whether the first method or the second method is active based on whether a corresponding method counter has exceeded a count threshold, reclaiming the native code associated with the first method and compiling byte code into native code associated with the second method in response to determining that the second method is active, and updating the method table for the first method. See Specification, paragraphs 0026-0035 and Figures 3-8.

In one embodiment, a machine-readable medium is disclosed. The machine-readable medium comprising instructions which, when executed, cause a machine to store native code associated with a first method and a second method within a native code space, and associate a reference to the first method and the second method in a method table. The instructions which, when executed, further cause the machine to determine whether the first method or the second method is invoked by detecting whether the first method or the second method correspond to the native code, and determine whether the native code space exceeds a threshold in response to an invocation of the first method and

the second method. The instructions which, when executed, further cause the machine to increment method counters each time the first method or the second method is invoked, wherein the method counters correspond to the first method and the second method. The instructions which, when executed, further cause the machine to unwind a stack to determine whether the first method or the second method is active based on whether a corresponding method counter has exceeded a count threshold, reclaim the native code associated with the first method and compiling byte code into native code associated with the second method in response to determining that the second method is active, and update the method table for the first method. See Specification, paragraphs 0026-0035 and Figures 3-8.

In one embodiment, a data processing system is disclosed. The system comprises a storage device and a processor coupled with the storage device. The processor is to process data and execute instructions. The processor is to store native code associated with a first method and a second method within a native code space, and associate a reference to the first method and the second method in a method table. The processor is further to determine whether the first method or the second method is invoked by detecting whether the first method or the second method correspond to the native code, and determine whether the native code space exceeds a threshold in response to an invocation of the first method and the second method. The processor is further to increment method counters each time the first method or the second method is invoked, wherein the method counters correspond to the first method and the second method. The processor is to unwind a stack to determine whether the first method or the second method is active based on whether a corresponding method counter has exceeded a count threshold, reclaim the native code associated with the first method and compiling byte

code into native code associated with the second method in response to determining that the second method is active, and update the method table for the first method. See Specification, paragraphs 0026-0035 and Figures 3-8.

VI. GROUND OF REJECTION

Claims 1-15 and 17-24 stand rejected under 35 U.S.C. § 102(e), as being anticipated by Ogasawara, U.S. Patent No. 6,671,877 (hereinafter “Ogasawara”).

No claims were indicated as allowable.

VII. ARGUMENT

- A. THE PENDING CLAIMS WERE IMPROPERLY REJECTED UNDER 35 U.S.C. § 102(E) BECAUSE OGASAWARA DOES NOT TEACH OR REASONABLY SUGGEST INCREMENTING METHOD COUNTERS EACH TIME THE FIRST METHOD OR THE SECOND METHOD IS INVOKED, WHEREIN THE METHOD COUNTERS CORRESPOND TO THE FIRST METHOD AND THE SECOND METHOD AND UNWINDING A STACK TO DETERMINE WHETHER THE FIRST METHOD OR THE SECOND METHOD IS ACTIVE BASED ON WHETHER A CORRESPONDING METHOD COUNTER HAS EXCEEDED A COUNT THRESHOLD

Claims 1-8 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Ogasawara. Appellant respectfully submits that Ogasawara does not teach or reasonably suggest the claimed invention for at least the reasons set forth below.

Claim 1, recites:

A method comprising:
storing native code associated with a first method and a second method within a native code space;
associating a reference to the first method and the second method in a method table;
determining whether the first method or the second method is invoked by detecting whether the first method or the second method correspond to the native code;
determining whether the native code space exceeds a threshold in response to an invocation of the first method or the second method;
incrementing method counters each time the first method or the second method is invoked, wherein the method counters correspond to the first method and the second method;
unwinding a stack to determine whether the first method or the second method is active based on whether a corresponding method counter has exceeded a count threshold;
reclaiming the native code associated with the first method and compiling byte code into native code associated with the second method in response to determining that the second method is active; and
updating the method table for the first method.
(emphasis added)

Ogasawara discloses calculating an activity degree of a method and using it to

decide an activity degree of each method and based on such an activity degree methods are discarded; however, Ogasawara does not teach or reasonably suggest determining whether the native code space exceeds a threshold in response to an invocation of a second method and reclaiming the native code. Instead, Ogasawara discloses calculating an activity degree of a method in order to determine when the method should be discarded.

For example, Ogasawara discloses “if a memory request of a JIT compiler cannot be met in a certain thread, the thread temporarily stops all the threads other than itself and locates active methods. A method for *calculating an activity degree of a method* is used to decide *the activity degree of each method*. Thereafter, *based on such an activity degree*, some or all of JITed codes of a non-active *method are discarded*”. Col 4, lines 8-14; emphasis added. The *activity degree of a method likely to be called early is raised according to how early it is called*. Thus, as to an *activity degree*, the earlier re-execution time of a method is, the larger its value becomes. Col. 3, lines 63-67; emphasis added. Ogasawara further discloses “a JIT compiler discards JITed code whose activity degree is lower, restarts thread execution, and continues compilation”. Col. 6, lines 65-67; emphasis added.

In contrast, claim 1, in pertinent part, recites “incrementing method counters each time the first method or the second method is invoked, wherein the method counters correspond to the first method and the second method [and] unwinding a stack to determine whether the first method or the second method is active based on whether a corresponding method counter has exceeded a count threshold.” (emphasis added).

Appellant submits that Ogasawara does not teach or reasonably suggest such a feature.

Ogasawara discloses “calculating an activity degree of a method” (col. 4, line 12);

however, nowhere does Ogasawara teach or reasonably suggest “incrementing method counters each time the first method or the second method is invoked, wherein the method counters correspond to the first method and the second method [and] unwinding a stack to determine whether the first method or the second method is active based on whether a corresponding method counter has exceeded a count threshold” as recited by claim 1. (emphasis added).

Appellant respectfully submits that claim 1 and its dependant claims are patentable over Ogasawara.

B. THE PENDING CLAIMS WERE IMPROPERLY REJECTED UNDER 35 U.S.C. § 102(E) BECAUSE OGASAWARA DOES NOT TEACH OR REASONABLY SUGGEST DETERMINING WHETHER THE FIRST METHOD OR THE SECOND METHOD IS INVOKED BY DETECTING WHETHER THE FIRST METHOD OR THE SECOND METHOD CORRESPOND TO THE NATIVE CODE AND DETERMINING WHETHER THE NATIVE CODE SPACE EXCEEDS A THRESHOLD IN RESPONSE TO AN INVOCATION OF THE FIRST METHOD OR THE SECOND METHOD

Claims 1-8 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Ogasawara. Appellant respectfully submits that Ogasawara does not teach or reasonably suggest the claimed invention for at least the reasons set forth below.

Claim 1, recites:

A method comprising:
storing native code associated with a first method and a second method within a native code space;
associating a reference to the first method and the second method in a method table;
determining whether the first method or the second method is invoked by detecting whether the first method or the second method correspond to the native code;
determining whether the native code space exceeds a threshold in response to an invocation of the first method or the second method;
incrementing method counters each time the first method or the second method is invoked, wherein the method counters correspond to the first method and the second method;
unwinding a stack to determine whether the first method or the second method is active based on whether a corresponding method counter has exceeded a count threshold;
reclaiming the native code associated with the first method and compiling byte code into native code associated with the second method in response to determining that the second method is active; and
updating the method table for the first method.
(emphasis added)

Continuing with the arguments as set forth above with respect to **Argument A**

(pages 8-10) Ogasawara further discloses “*if a memory request of a JIT compiler cannot*

be met in a certain thread, the thread temporarily stops all the threads other than itself and locates active methods.” Col. 4, lines 8-11; emphasis added. Ogasawara does not teach or reasonably suggest determining whether the first method or the second method is invoked by detecting whether the first method or the second method correspond to the native code [and] determining whether the native code space exceeds a threshold in response to an invocation of the first method or the second method” as recited by claim 1. (emphasis added).

Appellant respectfully submits that claim 1 and its dependant claims are patentable over Ogasawara.

- C. **THE PENDING CLAIMS WERE IMPROPERLY REJECTED UNDER 35 U.S.C. § 102(E) BECAUSE OGASAWARA DOES NOT TEACH OR REASONABLY SUGGEST INCREMENTING METHOD COUNTERS EACH TIME THE FIRST METHOD OR THE SECOND METHOD IS INVOKED, WHEREIN THE METHOD COUNTERS CORRESPOND TO THE FIRST METHOD AND THE SECOND METHOD AND UNWINDING A STACK TO DETERMINE WHETHER THE FIRST METHOD OR THE SECOND METHOD IS ACTIVE BASED ON WHETHER A CORRESPONDING METHOD COUNTER HAS EXCEEDED A COUNT THRESHOLD**

Claims 9-15 stand rejected under 35 U.S.C. § 102(e) as being anticipated by

Ogasawara.

Claim 9 contains limitations similar to those of claim 1. Accordingly, as previously discussed with regard to claim 1 in **Argument A** (pages 8-10), Appellant submits that claim 9 and its dependent claims are patentable over Ogasawara.

D. THE PENDING CLAIMS WERE IMPROPERLY REJECTED UNDER 35 U.S.C. § 102(E) BECAUSE OGASAWARA DOES NOT TEACH OR REASONABLY SUGGEST DETERMINING WHETHER THE FIRST METHOD OR THE SECOND METHOD IS INVOKED BY DETECTING WHETHER THE FIRST METHOD OR THE SECOND METHOD CORRESPOND TO THE NATIVE CODE AND DETERMINING WHETHER THE NATIVE CODE SPACE EXCEEDS A THRESHOLD IN RESPONSE TO AN INVOCATION OF THE FIRST METHOD OR THE SECOND METHOD

Claims 9-15 stand rejected under 35 U.S.C. § 102(e) as being anticipated by

Ogasawara.

Claim 9 contains limitations similar to those of claim 1. Accordingly, as previously discussed with regard to claim 1 in **Argument A and B** (pages 8-12), Appellant submits that claim 9 and its dependent claims are patentable over Ogasawara.

E. THE PENDING CLAIMS WERE IMPROPERLY REJECTED UNDER 35 U.S.C. § 102(E) BECAUSE OGASAWARA DOES NOT TEACH OR REASONABLY SUGGEST INCREMENTING METHOD COUNTERS EACH TIME THE FIRST METHOD OR THE SECOND METHOD IS INVOKED, WHEREIN THE METHOD COUNTERS CORRESPOND TO THE FIRST METHOD AND THE SECOND METHOD AND UNWINDING A STACK TO DETERMINE WHETHER THE FIRST METHOD OR THE SECOND METHOD IS ACTIVE BASED ON WHETHER A CORRESPONDING METHOD COUNTER HAS EXCEEDED A COUNT THRESHOLD

Claims 17-24 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Ogasawara.

Claim 17 contains limitations similar to those of claim 1. Accordingly, as previously discussed with regard to claim 1 in **Argument A** (pages 8-10), Appellant submits that claim 17 and its dependent claims are patentable over Ogasawara.

F. THE PENDING CLAIMS WERE IMPROPERLY REJECTED UNDER 35 U.S.C. § 102(E) BECAUSE OGASAWARA DOES NOT TEACH OR REASONABLY SUGGEST DETERMINING WHETHER THE FIRST METHOD OR THE SECOND METHOD IS INVOKED BY DETECTING WHETHER THE FIRST METHOD OR THE SECOND METHOD CORRESPOND TO THE NATIVE CODE AND DETERMINING WHETHER THE NATIVE CODE SPACE EXCEEDS A THRESHOLD IN RESPONSE TO AN INVOCATION OF THE FIRST METHOD OR THE SECOND METHOD

Claims 17-24 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Ogasawara.

Claim 17 contains limitations similar to those of claim 1. Accordingly, as previously discussed with regard to claim 1 in **Argument A and B** (pages 8-12), Appellant submits that claim 17 and its dependent claims are patentable over Ogasawara

VIII. CONCLUSION

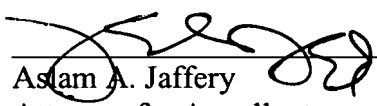
Appellant respectfully submits that all appealed claims in this application are patentable and were improperly rejected by the Examiner during prosecution before the United States Patent and Trademark Office. Appellant respectfully requests that the Board overrule the Examiner and direct allowance of the rejected claims.

This Brief is submitted with a check for \$500.00 to cover the appeal fee for one other than a small entity as specified in 37 C.F.R. §1.17(c). Please charge any shortages and credit any overpayments to our Deposit Account No. 02-2666.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN

Date: June 21, 2007


Aslam A. Jaffery
Attorney for Appellant
Registration Number: 51,841

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN
12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1030
(303)-740-1980

IX. APPENDIX OF CLAIMS (37 C.F.R. § 1.192(c)(7))



1. (Previously Presented) A method comprising:

storing native code associated with a first method and a second method within a

native code space;

associating a reference to the first method and the second method in a method

table;

determining whether the first method or the second method is invoked by

detecting whether the first method or the second method correspond to the

native code;

determining whether the native code space exceeds a threshold in response to an

invocation of the first method or the second method;

incrementing method counters each time the first method or the second method is

invoked, wherein the method counters correspond to the first method and

the second method;

unwinding a stack to determine whether the first method or the second method is

active based on whether a corresponding method counter has exceeded a

count threshold;

reclaiming the native code associated with the first method and compiling byte

code into native code associated with the second method in response to

determining that the second method is active; and

updating the method table for the first method.

2. (Previously Presented) The method of claim 2, wherein reclaiming the native code associated with the first method and compiling byte code into native code associated with the second method in response to the determination comprises reclaiming the native code associated with the first method in response to a determination that the native code space exceeds the threshold.
3. (Previously Presented) The method of claim 2, further comprising storing the native code associated with the second method within the native code space in response to the compilation.
4. (Previously Presented) The method of claim 2, further comprising:
invoking the first method following the reclamation; and
re-compiling byte code into the native code associated with the first method in response to the invocation of the first method.
5. (Previously Presented) The method of claim 2, wherein reclaiming the native code associated with the first method and compiling byte code into native code associated with the second method in response to the determination comprises compiling byte code into native code associated with the second method.
6. (Previously Presented) The method of claim 5, wherein compiling byte code into native code associated with the second method comprises compiling byte code

into native code associated with the second method utilizing a just-in-time compiler.

7. (Previously Presented) The method of claim 2, wherein reclaiming the native code associated with the first method and compiling byte code into native code associated with the second method in response to the determination comprises: determining whether the first method is active or inactive; and reclaiming the native code associated with the first method in response to a determination that the first method is inactive.
8. (Previously Presented) The method of claim 7, further comprising: reclaiming the native code associated with the first method and compiling byte code into native code associated with the second method in response to the determination further comprises determining whether the first method is hot or cold in response to a determination that the first method is inactive; and reclaiming the native code associated with the first method in response to a determination that the first method is inactive comprises reclaiming the native code associated with the first method in response to a determination that the first method is cold.

9. (Previously Presented) A machine-readable medium having stored thereon data representing sets of instructions which, when executed by a machine, cause the machine to:
- store native code associated with a first method and a second method within a native code space;
 - associate a reference to the first method and the second method in a method table;
 - determine whether the first method or the second method is invoked by detecting whether the first method or the second method correspond to the native code;
 - determine whether the native code space exceeds a threshold in response to an invocation of the first method and the second method;
 - increment method counters each time the first method or the second method is invoked, wherein the method counters correspond to the first method and the second method;
 - unwind a stack to determine whether the first method or the second method is active based on whether a corresponding method counter has exceeded a count threshold;
 - reclaim the native code associated with the first method and compiling byte code into native code associated with the second method in response to determining that the second method is active; and
 - update the method table for the first method.

10. (Previously Presented) The machine-readable medium of claim 9, wherein reclaiming the native code associated with the first method and compiling byte code into native code associated with the second method in response to the determination comprises reclaiming the native code associated with the first method in response to a determination that the native code space exceeds the threshold.
11. (Previously Presented) The machine-readable medium of claim 9, wherein the sets of instructions when further executed, cause the machine to perform operations comprising storing the native code associated with the second method within the native code space in response to the compilation.
12. (Previously Presented) The machine-readable medium of claim 9, wherein the sets of instructions when further executed, cause the machine to perform operations comprising invoking the first method following the reclamation; and re-compile byte code into the native code associated with the first method in response to the invocation of the first method.
13. (Previously Presented) The machine-readable medium of claim 9, wherein reclaiming the native code associated with the first method and compiling byte code into native code associated with the second method in response to the determination further cause the machine to compile byte code into native code associated with the second method.

14. (Previously Presented) The machine-readable medium of claim 13, wherein compiling byte code into native code associated with the second method further cause the machine to compile byte code into native code associated with the second method utilizing a just-in-time compiler.
15. (Previously Presented) The machine-readable medium of claim 9, wherein reclaiming the native code associated with the first method and compiling byte code into native code associated with the second method in response to the determination further cause the machine to:
- determine whether the first method is active or inactive; and
- reclaim the native code associated with the first method in response to a determination that the first method is inactive.
16. (Cancelled)
17. (Previously Presented) A data processing system comprising:
- a storage device; and
- a processor coupled with the storage device, the processor to process data and execute instructions, the processor to:
- store native code associated with a first method and a second method within a native code space;
- associate a reference to the first method and the second method in a method table;

determine whether the first method or the second method is invoked by detecting whether the first method or the second method correspond to the native code;

determine whether the native code space exceeds a threshold in response to an invocation of the first method and the second method;

increment method counters each time the first method or the second method is invoked, wherein the method counters correspond to the first method and the second method;

unwind a stack to determine whether the first method or the second method is active based on whether a corresponding method counter has exceeded a count threshold;

reclaim the native code associated with the first method and compiling byte code into native code associated with the second method in response to determining that the second method is active; and

update the method table for the first method.

18. (Previously Presented) The data processing system of claim 17, wherein when reclaiming the native code associated with the first method and compiling byte code into native code associated with the second method in response to the determination, the processor is further to reclaim the native code associated with the first method in response to a determination that the native code space exceeds the threshold.

19. (Previously Presented) The data processing system of claim 17, wherein the processor is further to store the native code associated with the second method within the native code space in response to the compilation.
20. (Previously Presented) The data processing system of claim 17, wherein the processor is further to invoke the first method following the reclamation; and re-compiling byte code into the native code associated with the first method in response to the invocation of the first method.
21. (Previously Presented) The data processing system of claim 17, wherein when reclaiming the native code associated with the first method and compiling byte code into native code associated with the second method in response to the determination, the processor is further to compile byte code into native code associated with the second method.
22. (Previously Presented) The data processing system of claim 21, wherein when compiling byte code into native code associated with the second method, the processor is further to compile byte code into native code associated with the second method utilizing a just-in-time compiler.
23. (Previously Presented) The data processing system of claim 17, wherein when reclaiming the native code associated with the first method and compiling byte

code into native code associated with the second method in response to the determination, the processor is further to:
determine whether the first method is active or inactive; and
reclaim the native code associated with the first method in response to a determination that the first method is inactive.

24. (Previously Presented) The data processing system of claim 23, wherein the processor is further to:
reclaim the native code associated with the first method and compiling byte code into native code associated with the second method in response to the determination further comprises determining whether the first method is hot or cold; and
reclaim the native code associated with the first method in response to a determination that the first method is inactive comprises reclaiming the native code associated with the first method in response to a determination that the first method is cold.

Claims 25-28 (Cancelled)

X. EVIDENCE APPENDIX

None.

XI. RELATED PROCEEDINGS APPENDIX

None.